A Lagrangian Perspective on Dual Propagation

Rasmus Kjær Høier*
Chalmers University of Technology

Christopher Zach
Chalmers University of Technology

Abstract

The search for "biologically plausible" learning algorithms has converged on the idea of representing gradients as activity differences. However, most approaches require a high degree of synchronization (distinct phases during learning) and introduce high computational overhead, which raises doubt regarding their biological plausibility as well as their potential usefulness for neuromorphic computing. Furthermore, they commonly rely on applying infinitesimal perturbations (nudges) to output units, which is impractical in noisy environments. Recently it has been shown that by modelling artificial neurons as dyads with two oppositely nudged compartments, it is possible for a fully local learning algorithm to bridge the performance gap to backpropagation, without requiring separate learning phases, while also being compatible with significant levels of nudging. However, the algorithm, called dual propagation, has the drawback that convergence of its inference method relies on symmetric nudging, which may be infeasible in biological and analog implementations. Starting from a modified version of LeCun's Lagrangian approach to backpropagation, we derive a slightly altered variant of dual propagation, which is robust to asymmetric nudging.

1 Introduction

Credit assignment using fully local alternatives to back-propagation is interesting both as potential models of biological learning as well as for their applicability for energy efficient analog neuromorphic computing (Kendall and Kumar, 2020; Yi et al., 2022). A pervasive idea in this field is the idea of representing error signals via activity differences, referred to as NGRAD (Neural Gradient Representation by Activity Differences) approaches (Lillicrap et al., 2020). However, a commonly overlooked issue in NGRAD approaches is the requirement of applying infinitesimal perturbations (or *nudging*) to output neurons in order to propagate error information through the network. This is problematic as analog and biological neural networks are inherently noisy, potentially causing the error signal to vanish if insufficient nudging is applied. In many local learning methods the output units are *positively* nudged to reduce a target loss, but utilizing additional negative nudging (output units increase a target loss) can be beneficial to improve accuracy (e.g. (Laborieux et al., 2021)).

The vanishing error signal problem is addressed by coupled learning (Stern et al., 2021), which proposes to replace the clamped output units of contrastive Hebbian learning with a convex combination of the label and the free phase outputs. Unfortunately, coupled learning has been shown to perform worse than equilibrium propagation on CIFAR10 and CIFAR100 (Scellier et al., 2023), and it does not necessarily approximate gradient descent on the output loss function (Stern et al., 2021). Holomorphic equilibrium propagation (Laborieux and Zenke, 2022, 2023) mitigates the need for infinitesimal nudging required in standard equilibrium propagation (Scellier and Bengio, 2017) at the cost of introducing complex-valued parameters. Whether this is a suitable approach for either biological or analog neural networks is an open question. Dual propagation (DP, Høier et al. (2023)),

*Correspondence to <hier@chalmers.se>

This work was supported by the National Supercomputer Centre at Linköping University and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

First Workshop on Machine Learning with New Compute Paradigms at NeurIPS 2023(MLNPCP 2023).

an algorithm similar in spirit to contrastive Hebbian learning, equilibrium propagation and coupled learning, is compatible with non-infinitesimal nudging by default. This method infers two sets of oppositely nudged and mutually tethered states simultaneously. However, utilization of symmetric nudging is a necessary condition for the convergence of its inference step.

Contributions DP is compatible with strong feedback and only requires a single inference phase, which are appealing features with regards to biological plausibility and potential applications to analog neuromorphic computing. However, the lack of convergence guarantees in the case of asymmetric nudging is clearly unsettling as exact symmetry is hard to realize outside of digital computers. Further,—unlike digital computers—neuromorphic, analog or otherwise highly distributed computing hardware typically performs continuous computations and runs asynchronously. Consequently, numerical stability of an energy-based inference and learning method is of essential importance. For this reason we derive an improved variant of dual propagation, which overcomes this strict symmetry requirement. In summary the contributions of this work are:

- Clarifying the connections between dual propagation and closely related methods (equilibrium propagation, contrastive Hebbian learning and lifted proximal operator machines).
- A new Lagrangian based derivation of dual propagation, which recovers the original dual propagation algorithm in the case of symmetric nudging, but leads to a slightly altered (and much more robust) method in the case of asymmetric nudging.
- Although this work is mainly theoretical in nature, we also experimentally verify that the improved DP method leads to stable learning in asymmetrically nudged settings, whereas the original DP approach yields poor or even diverging training behavior.

2 Related Work

CHL, EP and lifted networks In contrastive Hebbian learning (CHL) (Movellan, 1991; Xie and Seung, 2003) and equilibrium propagation (EP) (Scellier and Bengio, 2017) neuronal activations are found via an energy minimization procedure. Inference is carried out twice, once with and once without injecting label information at the output layer. CHL clamps the output units to the true targets, and EP nudges the output towards a lower loss. The difference between the activity in each of these two inference phases is used to represent neuronal error vectors. To ensure that inferred states represent the same local energy basin, this is typically done sequentially, e.g. the second inference phase is initialized with the solution found during the first phase.

Dual propagation (DP) is a closely related algorithm, in which each neuron has two intrinsic states corresponding to positively and negatively nudged compartments. The neural states in layer k are denoted $z_k^{\pm}$, where the superscripts indicates the direction of nudging. The neural activities and the error signals are represented by their weighted means and differences, respectively. The mean state is sent "upstream" to the next layer while the difference $z_k^+ - z_k^-$ is sent downstream to the preceding layer, where it is used to nudge $z_{k-1}^{\pm}$. This essentially "braids" the two inference phases and makes it possible to infer both states simultaneously. When the update sequence is chosen appropriately as little as two updates per neuron are sufficient, making the algorithm comparable to back-propagation in terms of runtime and more than 100X faster than CHL and EP.

Learning in CHL, EP and DP can be viewed as a bilevel optimization task over neural activities (the inner problem) and over weights (the outer problem). Expressed in terms of layerwise network potentials $E_k(z_k, z_{k-1}; W_{k-1})$, the training objectives of CHL, EP and DP can be expressed as:

$$\mathcal{L}^{CHL}(W) = \min_{\hat{z}} \max_{\check{z}} \sum\nolimits_{k=1}^{L} \beta^{k-L}(E_k(\hat{z}_k, \hat{z}_{k-1}) - E_k(\check{z}_k, \check{z}_{k-1})) \tag{1}$$

$$\mathcal{L}^{EP}(W) = \min_{\hat{z}} \max_{\check{z}} \ell(\hat{z}_L) + \tfrac{1}{\beta} \sum\nolimits_{k=1}^{L} (E_k(\hat{z}_k, \hat{z}_{k-1}) - E_k(\check{z}_k, \check{z}_{k-1})) \tag{2}$$

$$\mathcal{L}_{\alpha}^{DP}(W) = \min_{z^+} \max_{z^-} \alpha\ell(z_L^+) + \bar{\alpha}\ell(z_L^-) + \tfrac{1}{\beta} \sum\nolimits_{k=1}^{L} \left(E_k(z_k^+, \bar{z}_{k-1}) - E_k(z_k^-, \bar{z}_{k-1})\right), \tag{3}$$

where the dependence of E_k on the trainable parameters W is omitted for brevity. All activities for the input layer, $\hat{z}_0$, $\check{z}_0$ and $z_0^{\pm}$, are clamped to the network input x. The feedback (or nudging) parameter β determines the magnitude of perturbation introduced to the network potential by the target loss ℓ. β can vary between layers (or even between units), but for clarity we focus on the

usually sufficient setting of using a constant β for the entire network. $\alpha \in [0, 1]$ is a parameter specific to DP and steers the weighted average $\bar{z}_k := \alpha z_k^+ + (1 - \alpha) z_k^-$. A key difference between DP and the other two approaches is that the contrasted terms $E_k(z_k^+, \bar{z}_{k-1})$ and $E_k(z_k^-, \bar{z}_{k-1})$ both depend on this weighted average $\bar{z}_{k-1}$, which in practice means that z_k^+ and z_k^- are "tethered" to remain close to each other. In contrast to EP and CHL, dual propagation can infer both sets of states simultaneously, but the analysis of DP relies on choosing $\alpha = 1/2$.

Casting deep learning as optimization task over explicit activations and weights is the focus of a diverse set of back-propagation alternatives sometimes collectively referred to as *lifted neural networks* (Carreira-Perpinan and Wang, 2014; Askari et al., 2018; Gu et al., 2020; Li et al., 2019; Choromanska et al., 2019; Zach and Estellers, 2019; Høier and Zach, 2020). Although members of this group have different origins, they are algorithmically closely related to CHL and EP (Zach, 2021). Like DP, many of the lifted network approaches require a single inference phase, although activity inference still can be expensive, especially on digital hardware. The objective utilized in lifted proximal operator machines (Li et al., 2019; Zach, 2021) actually coincides with the one appearing in dual propagation when $\alpha = 1$. In this case z_k^- receives no feedback and can be expressed fully in terms of z_{k-1}^+ yielding a pure minimization objective:

$$\mathcal{L}^{LPOM}(W) = \min_{z^+} \ell(z_L^+) + \sum\nolimits_{k=1}^{L} \tfrac{1}{\beta} \left(E_k(z_k^+, z_{k-1}^+) - E_k(f_k(W_{k-1} z_{k-1}^+), z_{k-1}^+) \right) \tag{4}$$

Unfortunately it is not possible to infer z_k^+ in closed form when using $\mathcal{L}^{LPOM}$ and an iterative method (such as suitable fixed-point iterations) are required.

Difference target propagation Difference target propagation has emerged as a promising activity difference based learning algorithm. By applying a correction term to the targets computed by target propagation (LeCun, 1986; Bengio, 2014), DTP is able to better deal with non-invertible layers (Lee et al., 2015), which has been the Achilles heel of traditional target propagation. Recent work on difference target propagation has managed to close the performance gap compared to back-propagation in small CNNs (up to 5 hidden layers), by modifying the feedback weight learning scheme to establish stronger connections between back-propagation and the forward weight updates of DTP. The approach taken by Meulemans et al. (2020) produces a hybrid between gradient descent and Gauss-Newton updates, whereas the approach of Ernoult et al. (2022) aims to compute the same weight updates as backpropagation. The latter approach (Ernoult et al., 2022) yields (to our knowledge) the by far best results of any DTP implementation, but also introduces certain subtle architectural and algorithmic limitations. In particular the theoretical guarantees linking this flavor of DTP to BP, are only valid in the restricted setting of non-saturating and non-clamped backwards activations, and even then they are only valid in the case of single batch updates (details in App B).

Weak and strong feedback While a number of CHL-inspired learning methods for neural networks are shown to be equivalent to back-propagation when the feedback parameter β approaches zero (i.e. infinitesimal nudging takes place, as discussed in e.g. (Xie and Seung, 2003; Scellier and Bengio, 2017; Zach and Estellers, 2019; Zach, 2021)), practical implementations use a finite but small value for β, whose magnitude is further limited—either explicitly or implicitly. CHL implicitly introducing weak feedback via its layer discounting in order to approximate a feed-forward neural network, and both CHL and EP rely on weak feedback to stay in the same energy basin for the free and the clamped solutions. The iterative solver suggested for the LPOM model (Askari et al., 2018) also depends on sufficiently weak feedback to ensure convergence of the proposed fixed-point scheme to determine the neural activities. In contrast to these restrictions, the feedback parameter β in dual propagation is weakly constrained and its main effect is to influence the resulting finite difference approximation for activation function derivatives.

Compartmental perspective The contrastive, lifted and difference target propagation based models can all be interpreted as compartmental models (either compartments within a neuron or within a small neural circuitry). In EP and CHL, neurons need to store the activities of the two temporally distinct inference phases. In dual propagation and in LPOM, neurons are required to maintain the nudged internal states. Neurons in difference target propagation neurons are expected to have compartments for storing predictions, targets and possibly even noise perturbed states. Works such as (Guerguiev et al., 2017; Sacramento et al., 2018) explicitly focus on building biologically inspired compartmental neuron models, although these methods incur even higher computational costs by

also modelling spiking behaviour. Unlike the segregated dendrite model (Guerguiev et al., 2017), the dendritic cortical microcircuits (Sacramento et al., 2018) do not require distinct learning phases.

3 A Lagrangian Derivation of Dual Propagation

In this section we derive a variant of dual propagation, which turns out to be robust to asymmetric nudging (i.e. choosing $\alpha \in [0,1]$ not necessarily equal to $1/2$). Our starting point is a modified variant of LeCun's classic Lagrangian-based derivation of backpropagation (Lecun, 1988). We assume (i) that the activation functions f_k are all invertible (which can be relaxed), and (ii) that f_k has a symmetric derivative (i.e. $f_k'(x) = f_k'(x)^\top$). The second assumption clearly holds e.g. for element-wise activation functions. Our initial Lagrangian relaxation can now be stated as follows,

$$\mathcal{L}(W) = \min_z \max_\delta \ell(z_L) + \sum\nolimits_{k=1}^{L} \delta_k^\top \big(f_k^{-1}(z_k) - W_{k-1} z_{k-1}\big)\,. \tag{5}$$

Note that the multiplier δ_k corresponds to the constraint $f_k^{-1}(z_k) = W_{k-1}z_{k-1}$ (in contrast to the constraint $z_k = f_k(W_{k-1}z_{k-1})$ employed in (Lecun, 1988)). The main step is to reparamtrize z_k and δ_k in terms of z_k^+ and z_k^-,

$$z_k = \alpha z_k^+ + \bar\alpha z_k^- \qquad\qquad \delta_k = z_k^+ - z_k^- \tag{6}$$

for a parameter $\alpha \in [0,1]$ (and $\bar\alpha := 1-\alpha$). In the following we use the short-hand notations $\bar z_k := \alpha z_k^+ + \bar\alpha z_k^-$ and $a_k := W_{k-1}\bar z_{k-1}$.

Proposition 3.1. *Assume that the activation functions f_k, $k = 1,\dots,L-1$, are invertible, have symmetric derivatives and behave locally linear. Then the Lagrangian corresponding to the reparametrization in Eq. 6 is given by*

$$\mathcal{L}_\alpha^{DP+}(W) = \min_{z^+} \max_{z^-} \ell(\bar z_L) + \sum\nolimits_{k=1}^{L} (z_k^+ - z_k^-)^\top \big(f_k^{-1}(\bar z_k) - W_{k-1}\bar z_{k-1}\big)\,, \tag{7}$$

and the optimal $z_k^\pm$ in (7) *satisfy*

$$\begin{aligned} z_k^+ &\leftarrow f_k\big(W_{k-1}\bar z_{k-1} + \bar\alpha W_k^\top (z_{k+1}^+ - z_{k+1}^-)\big) \\ z_k^- &\leftarrow f_k\big(W_{k-1}\bar z_{k-1} - \alpha W_k^\top (z_{k+1}^+ - z_{k+1}^-)\big) \end{aligned} \tag{8}$$

for internal layers $k = 1,\dots,L-1$.

Proof. The first-order optimality conditions for z_k and δ_k ($k = 1,\dots,L-1$) in Eq. 5 are given by

$$\text{(I)}\quad 0 = (f_k^{-1})'(z_k)\delta_k - W_k^\top \delta_{k+1} \qquad\qquad \text{(II)}\quad 0 = f_k^{-1}(z_k) - W_{k-1}z_{k-1}. \tag{9}$$

Reparametrization in terms of $z_k^\pm$ (Eq. 6) and expanding (II) $+\,\alpha$(I) and (II) $-\,\bar\alpha$(I) yields

$$\begin{aligned} 0 = \;& f_k^{-1}(\alpha z_k^+ + \bar\alpha z_k^-) - a_k + \alpha (f_k^{-1})'(\bar z_k)^\top (z_k^+ - z_k^-) - \alpha W_k^\top (z_{k+1}^+ - z_{k+1}^-) \\ 0 = \;& -f_k^{-1}(\alpha z_k^+ + \bar\alpha z_k^-) + a_k + \bar\alpha (f_k^{-1})'(\bar z_k)^\top (z_k^+ - z_k^-) - \bar\alpha W_k^\top (z_{k+1}^+ - z_{k+1}^-), \end{aligned} \tag{10}$$

which are also the KKT conditions for $\mathcal{L}_\alpha^{DP+}$ in Eq. 7. It remains to manipulate these to obtain the desired update equations. Adding the equations above results in

$$0 = (f_k^{-1})'(\bar z_k)^\top (z_k^+ - z_k^-) - W_k^\top (z_{k+1}^+ - z_{k+1}^-). \tag{11}$$

Dual propagation is (via its use of finite differences) intrinsically linked to a (locally) linear assumption on f_k. Hence, we assume $f_k(a) = z_k^0 + D_k a + O(\|a - a_k\|^2)$ with $z_k^0 = f_k(a_k) - D_k a_k$ and $D_k = f_k'(a_k)$. The local linear assumption allows us to neglect the higher order terms. Consequently we also assume that f_k^{-1} is locally linear and therefore $(f_k^{-1})'(\bar z_k) \approx D_k^{-1}$ is independent of $\bar z_k$. Hence, we arrive at

$$0 \approx D_k^{-\top}(z_k^+ - z_k^-) - W_k^\top (z_{k+1}^+ - z_{k+1}^-) \iff z_k^+ - z_k^- \approx D_k^\top W_k^\top (z_{k+1}^+ - z_{k+1}^-). \tag{12}$$

We insert this into the second of the necessary optimality conditions (10) and obtain

$$\begin{aligned} 0 &= f_k^{-1}(\alpha z_k^+ + \bar\alpha z_k^-) - a_k = f_k^{-1}\big(z_k^- + \alpha(z_k^+ - z_k^-)\big) - a_k \\ &= f_k^{-1}\big(z_k^- + \alpha D_k^\top W_k^\top (z_{k+1}^+ - z_{k+1}^-)\big) - a_k = D_k^{-1}\big(z_k^- - z_k^0 + \alpha D_k^\top W_k^\top (z_{k+1}^+ - z_{k+1}^-)\big) - a_k. \end{aligned}$$

The last line is equivalent to

$$z_k^- = D_k a_k + z_k^0 - \alpha D_k^\top W_k^\top (z_{k+1}^+ - z_{k+1}^-) \approx f_k\big(W_{k-1}\bar{z}_{k-1} - \alpha W_k^\top (z_{k+1}^+ - z_{k+1}^-)\big)\,. \quad (13)$$

Analogously, $z_k^+ \approx f_k(W_{k-1}\bar{z}_{k-1} + \bar{\alpha} W_k^\top (z_{k+1}^+ - z_{k+1}^-))$. In summary we obtain the update rules shown in (8). □

For the output layer activities $z_L^\pm$ we absorb f_L into the target loss (if necessary) and therefore solve

$$\min_{z_L^+} \max_{z_L^-} \ell(\bar{z}_L) + (z_L^+ - z_L^-)^\top (\bar{z}_L - a_L) \quad (14)$$

with corresponding necessary optimality conditions

$$\alpha \ell'(\bar{z}_L) + \bar{z}_L - a_L + \alpha(z_L^+ - z_L^-) = 0 \qquad \bar{\alpha}\ell'(\bar{z}_L) - \bar{z}_L + a_L + \bar{\alpha}(z_L^+ - z_L^-) = 0. \quad (15)$$

Adding these equations reveals $\ell'(\bar{z}_L) + z_L^+ - z_L^- = 0$. Inserting this in any of the equations in (15) also implies $\bar{z}_L = a_L = W_{L-1}\bar{z}_{L-1}$. Thus, by combining these relations and with $g := \ell'(W_{L-1}\bar{z}_{L-1})$, the updates for $z_L^\pm$ are given by

$$z_L^+ \leftarrow W_{L-1}\bar{z}_{L-1} - \bar{\alpha} g \qquad\qquad z_L^- \leftarrow W_{L-1}\bar{z}_{L-1} + \alpha g. \quad (16)$$

For the β-weighted least-squares loss, $\ell(z_L) = \frac{\beta}{2}\|z_L - y\|^2$, the above updates reduce to $z_L^+ \leftarrow a_L - \bar{\alpha}\beta(a_L - y)$ and $z_L^- \leftarrow a_L + \alpha\beta(a_L - y)$.

Relation to the original dual propagation The update equations in (8) coincide with the original dual propagation rules if $\alpha = 1/2$ (Høier et al., 2023), although the underlying objectives $\mathcal{L}_\alpha^{DP+}$ (7) and $\mathcal{L}_\alpha^{DP}$ are fundamentally different. If $\alpha \neq 1/2$, then α and $\bar{\alpha}$ switch places w.r.t. the error signals (but not in the definition of $\bar{z}$) compared to the original dual propagation method.[2] The updates in (8) for $\alpha = 0$ correspond to an algorithm called "Fenchel back-propagation" discussed in (Zach, 2021).

Both the objective of the original dual propagation (3) and the objective of the improved variant (7) can be expressed in a general contrastive form, but the underlying potentials E_k are somewhat different,

$$\text{New DP: } E_k(z_k^\pm, \bar{z}_{k-1}) - E_k(z_k^\mp, \bar{z}_{k-1}) = (z_k^+ - z_k^-)^\top \nabla G_k(\bar{z}_k) - (z_k^+ - z_k^-)^\top W_{k-1}\bar{z}_{k-1} \quad (17)$$

$$\text{Orig. DP: } E_k(z_k^+, \bar{z}_{k-1}) - E_k(z_k^-, \bar{z}_{k-1}) = G_k(z_k^+) - G_k(z_k^-) - (z_k^+ - z_k^-)^\top W_{k-1}\bar{z}_{k-1}. \quad (18)$$

Here G_k is a convex mapping ensuring that $\arg\min_{z_k} E_k(z_k, z_{k-1})$ provides the desired activation function f_k. The relation between f_k and G_k is given by $f_k = \nabla G_k^*$ and $f_k^{-1} = \nabla G_k$ (where G_k^* is the convex conjugate of G_k). Activations function induced by G_k have automatically symmetric Jacobians as $f_k' = \nabla^2 G_k^*$ under mild assumptions.

Non-invertible activation function If f_k is not invertible at the linearization point (such as the softmax function), then D_k is singular and the constraint $f_k^{-1}(z_k) = W_{k-1}z_{k-1}$ is converted into a constraint that $D_k^+ z_k$ is restricted to a linear subspace,

$$D_k^+(z_k - z_k^0) = W_{k-1}z_{k-1} + N_k v_k, \quad (19)$$

where N_k spans the null space of D_k and v_k is an arbitrary vector. Going through the similar steps as above leads to the same updates for $z_k^\pm$.

Starting from Lecun's Lagrangian If we start from the more natural Lagrangian

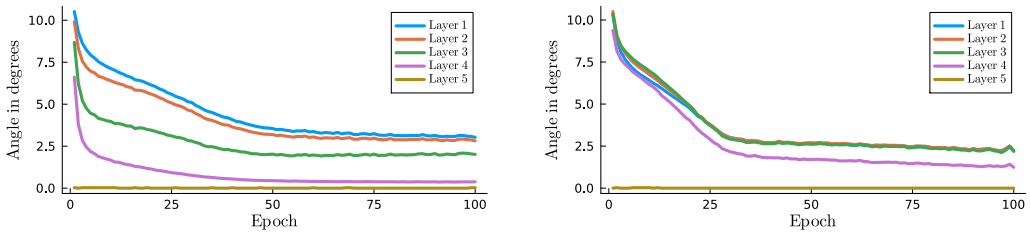
$$\mathcal{L}(W) = \min_z \max_\delta \ell(z_L) + \sum\nolimits_{k=1}^{L} \delta_k^\top (z_k - f_k(W_{k-1}z_{k-1}))\,, \quad (20)$$

instead from (5), then the step in (13) is not possible and the back-propagated signal $z_{k+1}^- - z_{k+1}^+$ cannot be moved inside the activation function f_k. The update equations for $z_k^\pm$ are of the less-convenient form

$$\begin{aligned} z_k^+ &\leftarrow f_k(W_{k-1}\bar{z}_{k-1}) + \bar{\alpha} W_k^\top f_{k+1}'(\bar{z}_k)(z_{k+1}^- - z_{k+1}^+) \\ z_k^- &\leftarrow f_k(W_{k-1}\bar{z}_{k-1}) - \alpha W_k^\top f_{k+1}'(\bar{z}_k)(z_{k+1}^- - z_{k+1}^+), \end{aligned} \quad (21)$$

which still require derivatives of the activation functions, which may be problematic for analog implementations.

[2]This partial exchange of roles of α and $\bar{\alpha}$ is somewhat analogous to the observation that e.g. forward differences in the primal domain become backward differences in the adjoint.



(a) $\alpha = 0$ (b) $\alpha = 1$

Figure 1: Alignment between the parameter updates obtained with back-propagation and with the improved DP variant (using 30 inference iterations and asymmetric nudging with $\alpha \in \{0, 1\}$). Results are averaged over five random seeds.

4 Numerical Validation

The original dual propagation and the improved variant differ in the case $\alpha \in [0, 1]$, $\alpha \neq 1/2$, with the boundary cases $\alpha \in \{0, 1\}$ being of particular interest. The most efficient way to implement dual propagation is by sequentially updating neurons in every layer from input to output and then from output to input (akin to a forward and a backward traversal, which are nevertheless part of the sole inference phase using the same dynamics) before performing a weight update. However, to show the unstable behaviour of the original dual propagation formulation in the case of asymmetric nudging we instead perform repeated inference passes through the layers (which also better matches a continuous/asynchronous compute model). Each pass (or iteration) corresponds to updating all neurons from input to output layer and back.

The results of the experiments are summarized in Tab 1, where we trained a 784-1000(×4)-10 MLP with ReLU activation functions on MNIST using the least-squares loss. The nudging strength β is 1/2, which is also compatible with the original DP method. We observe that inference in the original variant of DP diverges when applying asymmetric nudging and multiple inference iterations. This is not surprising as inference for dual propagation is only guaranteed to converge in the symmetric setting. The new variant of DP on the other hand yields stable inference results in all cases, and further shows good alignment with back-propagation gradients in both of the fully asymmetric settings ($\alpha \in \{0, 1\}$) as shown in Fig 1.

Table 1: Mean test accuracy in percent for the original and the improved dual propagation methods using $\alpha \in \{0, 1\}$. X indicates that the particular experiment did not converge. Results are averaged over five random seeds.

	$\alpha = 0$		$\alpha = 1$	
Iterations	Improved DP	Original DP	Improved DP	Original DP
1	98.33 ± 0.10	98.50 ± 0.08	98.43 ± 0.08	97.92 ± 0.11
30	98.36 ± 0.03	X	98.46 ± 0.09	X

5 Conclusion

Fully local activity difference based learning algorithms are essential for achieving on-device training of neuromorphic hardware. However, the majority of works rely on applying weak nudging to output neurons in order to propagate error signals, which is problematic in noisy hardware. The original dual propagation formulation overcomes this issue but also relies on symmetric nudging, which may itself be too strict a requirement in noisy hardware. In this work we present a novel Lagrangian based derivation of an improved instance of dual propagation, which recovers the exact dynamics of the original dual propagation formulation in the case of symmetric nudging ($\alpha = 1/2$). In the case of asymmetric nudging ($\alpha \in [0, 1]$, $\alpha \neq 1/2$) the new formulation leads to slightly modified and importantly stable inference dynamics. In experiments we verify that the new formulation leads to stable dynamics in the case of fully asymmetric nudging ($\alpha = 1$ and $\alpha = 0$) and repeated inference where the original formulation struggles.

References

A. Askari, G. Negiar, R. Sambharya, and L. E. Ghaoui. Lifted neural networks. *arXiv preprint arXiv:1805.01532*, 2018.

Y. Bengio. How auto-encoders could provide credit assignment in deep networks via target propagation. *arXiv preprint arXiv:1407.7906*, 2014.

M. Carreira-Perpinan and W. Wang. Distributed optimization of deeply nested systems. In *Artificial Intelligence and Statistics*, pages 10–19, 2014.

A. Choromanska, B. Cowen, S. Kumaravel, R. Luss, M. Rigotti, I. Rish, P. Diachille, V. Gurev, B. Kingsbury, R. Tejwani, et al. Beyond backprop: Online alternating minimization with auxiliary variables. In *International Conference on Machine Learning*, pages 1193–1202. PMLR, 2019.

M. M. Ernoult, F. Normandin, A. Moudgil, S. Spinney, E. Belilovsky, I. Rish, B. Richards, and Y. Bengio. Towards scaling difference target propagation by learning backprop targets. In *International Conference on Machine Learning*, pages 5968–5987. PMLR, 2022.

F. Gu, A. Askari, and L. El Ghaoui. Fenchel lifted networks: A lagrange relaxation of neural network training. In *International Conference on Artificial Intelligence and Statistics*, pages 3362–3371. PMLR, 2020.

J. Guerguiev, T. P. Lillicrap, and B. A. Richards. Towards deep learning with segregated dendrites. *eLife*, 6:e22901, dec 2017. ISSN 2050-084X. doi: 10.7554/eLife.22901.

R. Høier and C. Zach. Lifted regression/reconstruction networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020.

R. Høier, D. Staudt, and C. Zach. Dual propagation: Accelerating contrastive hebbian learning with dyadic neurons. In *International Conference on Machine Learning*, 2023. URL `https://icml.cc/virtual/2023/poster/23795`.

J. D. Kendall and S. Kumar. The building blocks of a brain-inspired computer. *Applied Physics Reviews*, 7(1):011305, 2020.

A. Laborieux and F. Zenke. Holomorphic equilibrium propagation computes exact gradients through finite size oscillations. *Advances in Neural Information Processing Systems*, 35:12950–12963, 2022.

A. Laborieux and F. Zenke. Improving equilibrium propagation without weight symmetry through jacobian homeostasis. *arXiv preprint arXiv:2309.02214*, 2023.

A. Laborieux, M. Ernoult, B. Scellier, Y. Bengio, J. Grollier, and D. Querlioz. Scaling equilibrium propagation to deep convnets by drastically reducing its gradient estimator bias. *Frontiers in neuroscience*, 15:129, 2021.

Y. LeCun. Learning process in an asymmetric threshold network. In *Disordered systems and biological organization*, pages 233–240. Springer, 1986.

Y. Lecun. A theoretical framework for back-propagation. In *Proceedings of the 1988 Connectionist Models Summer School, CMU, Pittsburg, PA*, pages 21–28. Morgan Kaufmann, 1988.

D.-H. Lee, S. Zhang, A. Fischer, and Y. Bengio. Difference target propagation. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 498–515. Springer, 2015.

J. Li, C. Fang, and Z. Lin. Lifted proximal operator machines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4181–4188, 2019.

T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

A. Meulemans, F. Carzaniga, J. Suykens, J. Sacramento, and B. F. Grewe. A theoretical framework for target propagation. *Advances in Neural Information Processing Systems*, 33:20024–20036, 2020.

J. R. Movellan. Contrastive hebbian learning in the continuous hopfield model. In *Connectionist models*, pages 10–17. Elsevier, 1991.

J. Sacramento, R. Ponte Costa, Y. Bengio, and W. Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in neural information processing systems*, 31, 2018.

B. Scellier and Y. Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

B. Scellier, M. Ernoult, J. Kendall, and S. Kumar. Energy-based learning algorithms: A comparative study. In *ICML Workshop on Localized Learning (LLW)*, 2023. URL `https://openreview.net/forum?id=VLszAxAFGs`.

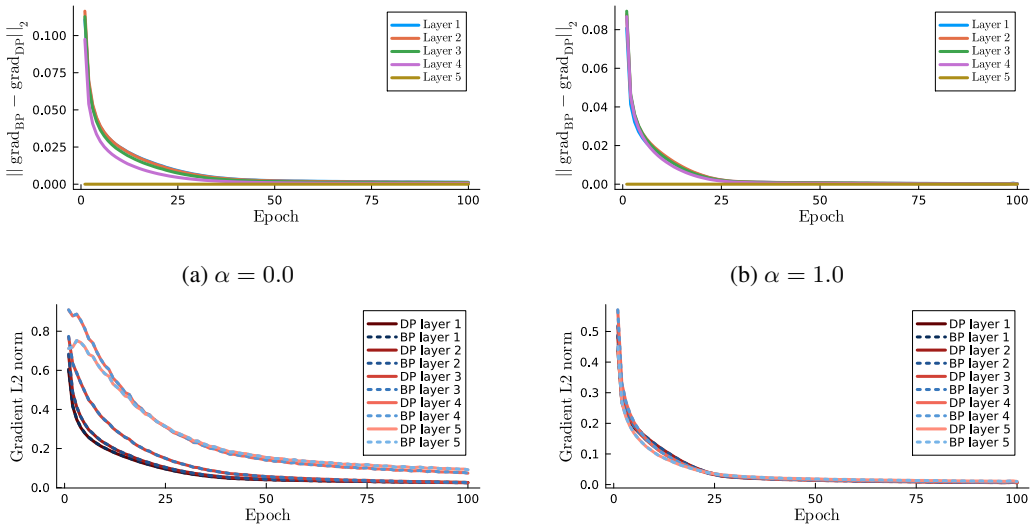
M. Stern, D. Hexner, J. W. Rocks, and A. J. Liu. Supervised learning in physical networks: From machine learning to learning machines. *Physical Review X*, 11(2):021045, 2021.

X. Xie and H. S. Seung. Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, 15(2):441–454, 2003.

S.-i. Yi, J. D. Kendall, R. S. Williams, and S. Kumar. Activity-difference training of deep neural networks using memristor crossbars. *Nature Electronics*, pages 1–7, 2022.

C. Zach. Bilevel programs meet deep learning: A unifying view on inference learning methods. *arXiv preprint arXiv:2105.07231*, 2021.

C. Zach and V. Estellers. Contrastive learning for lifted networks. In K. Sidorov and Y. Hicks, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 163.1–163.12. BMVA Press, 9 2019. doi: 10.5244/C.33.163.



(a) $\alpha = 0.0$ (b) $\alpha = 1.0$

(c) $\alpha = 0.0$ (d) $\alpha = 1.0$

Figure 2: **Top:** L2 norm of difference between BP and DP gradients and **Bottom:** L2 norms of BP and DP gradients when using 30 inference iterations and asymmetric nudging ($\alpha = 0.0$ and $\alpha = 1.0$). Results are averaged over five random seeds.

A Additional Results

Fig. 2 illustrate additional statistics on the alignment of back-propagation induced gradients (w.r.t. the network parameters) and the ones obtained by the proposed DP improvement.

B Implicit Constraints for Jacobian Matching in Difference Target Propagation

The proof of difference target propagation's ability to compute backpropagation gradients in (Ernoult et al., 2022) relies on the Jacobian Matching Condition (JMC) being fullfilled, which we find is only possible under specific conditions.

Assuming a fully connected architecture, omitting biases for simplicity and mirroring the notation (Ernoult et al., 2022) we have the vector of feed-forward activations in layer n as $s^n = F^{n-1}(s^{n-1}) = f^{n-1}(\theta^{n-1}s^{n-1})$, where f^{n-1} and θ^{n-1} respectively are the activation function and weights associated with the mapping from layer $n-1$ to layer n. The backwards mapping $G^n(s^{n+1})$ is used when computing targets and performs the operations $G^n(s^{n+1}) = \omega^n g^n(s^{n+1})$. Here g^n and ω^n respectively are the activation function and weights associated with the backwards mapping from layer $n+1$ to layer n.

The Jacobian matching condition is:

$$\begin{aligned} (\partial_{s^n} F^n(s^n))^\top &= \partial_{s^{n+1}} G^n(s^{n+1}) \\ \Leftrightarrow \\ Df^n\theta^n &= Dg^n(\omega^n)^\top, \end{aligned} \tag{22}$$

where Df^n is shorthand notation denoting the diagonal matrix containing the derivatives of $f^n(\theta^n s^n)$ on the diagonal and zero everywhere else. Analogously Dg^n denotes the diagonal matrix containing the derivatives of $g^n(s^{n+1})$.

In the paper it is stated that the JMC is guaranteed to be satisfied when the following objective is minimized with respect to ω:

$$\hat{\mathcal{L}}^n_\omega = -\tfrac{1}{\sigma^2}\epsilon^T(r^n_\epsilon - s^n) + \tfrac{1}{2\sigma^2}||r^n_\eta - s^n||^2 \tag{23}$$

It is proven that (in the limit as noise goes to zero) gradient descent on Eq 23 with respect to the backwards parameters ω is equivalent to gradient descent on the following objective

$$L_{\omega}^{n} = \frac{1}{2}||\partial_{s^n} F^{n\top} - \partial_{s^{n+1}} G^n||_F^2 = \frac{1}{2}||Df^n\theta^n - Dg^n(\omega^n)^\top||^2, \tag{24}$$

Furthermore, it is clear that if the minimization procedure yields $L_{\omega}^{n} = 0$, then the JMC is fullfilled and gradient matching is achieved, but this is not possible for all choices of g^n. In particular in the case of saturating or clamping backwards activations. The Jacobian matching condition the matrix $\partial_{s^{n+1}} G^n = Dg^n(\omega^n)$ can have entire rows of zeros, which can make it impossible for the loss to reach zero value, by simply minimizing with respect to ω^n. This issue can be avoided by using linear backwards activations, as done in the paper (Ernoult et al., 2022). However, due to the presence of the datapoint dependent Df^n in the JMC it will in general be necessary to solve for ω^n on a per datapoint basis. I.e. the proposed procedure will not be able to perfectly align DTP gradient estimates to backprop gradients unless the batchsize is set equal to 1.